46th AIAA Aerospace Sciences Meeting and Exhibit
7 - 10 January 2008, Reno, Nevada

AIAA 2008-783

A Study of Viscous Flux Formulations for an Implicit P-Multigrid Spectral Volume Navier Stokes Solver

R. Kannan*, Y. Sun†, and Z.J. Wang‡
Department of Aerospace Engineering, Iowa State University, Ames, IA 50011

In this paper, we improve the Navier-Stokes flow solver developed by Sun et al based on the spectral volume method in the following two aspects: the development of a more efficient implicit/p-multigrid solution approach, and the use of a new viscous flux formula. An implicit preconditioned LU-SGS p-multigrid method developed for the spectral difference (SD) Euler solver by Liang et al is adopted here. In the original SV solver, the viscous flux was computed with a local discontinuous Galerkin-type approach. In this study, a penalty approach based on the first method of Bassi and Rebay is suggested and tested for both the Laplace and Navier-Stokes equations. The second method of Bassi and Rebay is also tested for the Laplace equation. Their convergence properties are studied in the context of the BLU-SGS approach. Fourier analysis revealed some interesting advantages for the penalty method over the LDG method. A convergence speedup of up to 2 orders is obtained with the implicit method. The convergence was further enhanced by employing a p-multigrid algorithm. The numerical results are very promising and indicate that the approach has a great potential for 3D flow problems.

Introduction

The spectral volume (SV) method was originally developed by Wang, Liu et al for hyperbolic conservation laws on unstructured grids [1-5]. The spectral volume method is a subset of the more generic Godunov type finite volume method, which is the starting block for the development of a plethora of methods such as the k-exact finite volume [6,7], the essentially non-oscillatory (ENO) [8,9], and total variation diminishing (TVD) [10] schemes. The spectral volume method can be viewed as an extension of the Godunov method to higher order by adding more degrees-of-freedom (DOFs) in the form of subcells in each cell (simplex). The simplex is referred to as a spectral volume (SV) and the subcells are referred to as control volumes (CV). As in the finite volume method, the unknowns (or DOFs) are the subcell-averaged solutions. A finite volume procedure is employed to update the DOFs.

Recently Sun et al [11] implemented the SV method for the Navier Stokes equations. The local discontinuous Galerkin (LDG) [12] approach was used in the SV method to discretize the viscous fluxes. The LDG approach alternates between the right and the left directions for obtaining the CV averaged gradient and the residual respectively. More recent numerical tests indicated that the computational results using LDG are somewhat dependent on how the faces are oriented, especially for unstructured and non uniform grids. In this paper, we test another technique to discretize the viscous fluxes. The new procedure is a penalty-like method based on the algorithm developed by Bassi and Rebay [13]. This procedure is more symmetrical than LDG and hence is better suited for unstructured and non uniform grids. Fourier analysis revealed better accuracy, higher time step limit (for explicit simulations) and faster convergence for the penalty approach when compared to LDG.

The convergence properties of high order methods like SV, SD and the discontinuous Galerkin (DG) are generally poor with explicit time marching approaches. In this paper, we adopt an efficient preconditioned implicit LU-SGS approach. The original LU-SGS approach was developed by Yoon and Jameson on structured grids [14], and later improved upon by Jameson and Caughey [15], Chen & Wang [16] using an element Jacobi preconditioner. It was used by Liang et al [17] to solve the Euler equations with a p-multigrid algorithm for the SD method. We are able to achieve significant speedup using the implicit approach.

---

*Graduate Research Assistant
†Post Doctoral Assistant
‡Currently Associate Professor of Aerospace Engineering, Iowa State University, Associate Fellow of AIAA

In addition, we also use a p-multigrid algorithm to enhance the convergence. The p-multigrid method employs smoothing operators which hasten the convergence by using the coarse levels constructed by reducing the order of the interpolation polynomial p. This method was initially proposed by Ronquist and Patera [18], and extended by Maday and Munoz [19]. The acceleration of the convergence by the p-multigrid algorithm on Euler equations was demonstrated by Bassi and Rebay [20], Fidkowski et al [21], Nastase and Mavriplis [22] and Luo et al. [23] with the DG method and Liang et al [17] for the SD method.

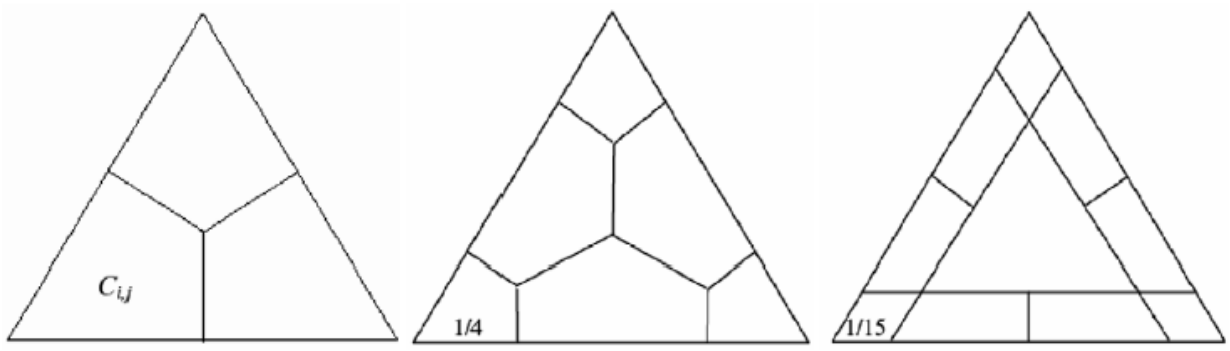
The paper is organized as follows. In the next section, we review the basics of the SV method. The new penalty formulation will be presented, together with a detailed linear analysis in section 3. The results are then compared with the LDG scheme. The implicit LUSGS scheme is discussed in section 4. In Section 5, we focus on the nonlinear FAS p-multigrid method. Section 6 deals with the different test cases conducted during this study. Finally conclusions from this study are summarized in Section 7.

Basics of the Spectral volume method

Consider the general conservation equation

$$\frac{\partial Q}{\partial t} + \frac{\partial (f_i(Q) - f_v(Q))}{\partial x} + \frac{\partial (g_i(Q) - g_v(Q))}{\partial y} = 0 \tag{1.1}$$

in domain Ω with the appropriate initial and boundary conditions . In (1.1), x and y are the Cartesian coordinates and $(x,y) \in \Omega$, $t \in [0,T]$ denotes time, Q is the vector of conserved variables, and f_i and g_i are the inviscid fluxes in the x and y directions, respectively. f_v and g_v are the viscous fluxes in the x and y directions, respectively. Domain Ω is discretized into I non-overlapping triangular (2D) cells. In the SV method, the simplex grid cells are called SVs, denoted S_i , which are further partitioned into CVs, denoted C_{ij} , which depend on the degree of the polynomial reconstruction. Figure 1 shows linear, quadratic and cubic partitions in 2D.



a **b** **c**

Figure 1. Partitions of a triangular SV . Linear, quadratic and cubic reconstructions are shown in a, b and c respectively.

We need N unknown control volume solution averages (or DOFs). N is calculated using the below formula

$$N = \frac{(m+1)(m+2)}{2} \tag{1.2}$$

where m is the degrees of the polynomial, constructed using the CV solution averages. The CV averaged conserved variable for C_{ij} is defined as

$$\overline{Q}_{i,j} = \frac{1}{V_{i,j}} \int_{C_{i,j}} Q dV \text{ , j=1,…N, i=1,…I,} \tag{1.3}$$

where $V_{i,j}$ is the volume of C_{ij}. Given the CV averaged conserved variables, a m^{th} degree polynomial can be constructed such that it is $(m+1)^{th}$ order approximation to Q. In other words, we can write the polynomial as

$$p_i(x,y) = \sum_{j=1}^{N} L_j(x,y) \overline{Q}_{i,j} \tag{1.4}$$

where the shape functions $L_j(x,y)$ satisfy

$$\frac{1}{V_{i,j}}\int_{C_{i,j}} L_n(x,y)dV = \delta_{j,n} \tag{1.5}$$

Equation 1.1 is integrated over the C_{ij}. This results in the below equation

$$\frac{\partial \overline{Q}}{\partial t} + \frac{1}{V_{i,j}}\sum_{r=1}^{K}\int_{Ar}(\vec{F}.\vec{n})dA = 0 \tag{1.6}$$

where $\vec{F} = (f_i - f_v, g_i - g_v)$ where A_r represents the r^{th} face of C_{ij}, $\vec{n}$ is the outward unit normal vector of A_r and K is the number of faces in C_{ij}. The surface integration on each face is done using a $(m+1)^{th}$ order accurate Gauss quadrature formula. The fluxes are discontinuous across the SV interfaces. The inviscid fluxes are handled using a numerical Riemann flux like the Rusanov flux [24], the Roe flux [25] or AUSM flux [26]. The handling of the viscous fluxes is discussed in the next section.

Spectral volume formulation for the diffusion equation

A. Discretization of viscous fluxes

The following 2D diffusion equation is considered first in domain Ω with the appropriate initial and boundary conditions

$$\frac{\partial u}{\partial t} - \nabla\cdot(\mu\nabla u) = 0 \tag{2.1}$$

where μ is a positive diffusion coefficient. We define an auxiliary variable

$$q = \nabla u \tag{2.2}$$

Equation 2.1 then becomes

$$\frac{\partial u}{\partial t} - \nabla\cdot(\mu q) = 0 \tag{2.3}$$

Using the Gauss-divergence theorem, we obtain

$$\overline{q_{ij}}V_{ij} = \sum_{r=1}^{K}\int_{Ar} u\cdot n dA \tag{2.4}$$

$$\frac{d\overline{u}_{ij}}{dt}V_{ij} - \sum_{r=1}^{K}\int_{A_r}\mu q\cdot n dA = 0 \tag{2.5}$$

where $\overline{q}_{ij}$ and $\overline{u}_{ij}$ are the CV averaged gradient and solution in C_{ij}, K is the number of faces in C_{ij}, A_r represents the rth face of C_{ij} and n is the unit surface normal vector. As the solution u is cell-wise continuous, u and q at SV boundaries are replaced by numerical fluxes $\underline{q}$ and $\underline{u}$. The above equations thus become

$$\overline{q_{ij}}V_{ij} = \sum_{r=1}^{K}\int_{Ar}\underline{u}\cdot n dA \tag{2.6}$$

$$\frac{d\overline{u}_{ij}}{dt}V_{ij} - \sum_{r=1}^{K}\int \mu\underline{q}\cdot n dA = 0\,. \tag{2.7}$$

The commonly used approach for obtaining the numerical fluxes is the LDG approach. In this approach, the numerical fluxes are defined by alternating the direction in the following manner [12]

$$\underline{u} = u_L \tag{2.8}$$

$$\underline{q} = q_R \tag{2.9}$$

A1. Penalty approach

It can be seen (from equations 2.8 and 2.9) that LDG is inherently unsymmetric. A symmetric approach was given by Bassi and Rebay [13], in which the numerical fluxes are defined by

$$\underline{u} = 0.5*(u_R + u_L) \tag{2.10}$$

$$\underline{q} = 0.5*(q_R + q_L) \tag{2.11}$$

Analysis by Brezzi et al [27] showed that the approach may be unstable in some situations. In this paper, we suggest the following the penalty approach to obtain the numerical fluxes:

$$\underline{u} = 0.5*(u_R + u_L) \tag{2.10}$$

$$\underline{q} = 0.5*(q_R + q_L) + (u_R - u_L)n\frac{A_r}{V_{ij}} \tag{2.11}$$

A2. Second approach of Bassi and Rebay(BR2)

The penalty approach is definitely more symmetrical than LDG. However, the implementation results in a non-compact stencil. We now give the formulation of BR2 [28], which is compact. The BR2 formulation is slightly different from the above two methods. Two different gradients are evaluated for residual computation:

I. The first gradient is referred to as the self gradient. It is computed using the solution from the cell in consideration. In other words, there is no contribution from the neighboring cells. This gradient is utilized for computing the viscous fluxes through the SV boundary faces.

II. The second gradient is referred to as the averaged gradient. It is computed by using the averaged solution obtained from the left and the right sides of the SV interfaces. This gradient is used for the internal faces of the SV

BR2 uses the concept of lifting functions. These lifting functions, $\vec{r}^{+}$ and $\vec{r}^{-}$ (which are actually polynomials) can be viewed as corrections to the self gradient. The CV averaged lifting functions $\overline{\vec{r}^{+}}$ and $\overline{\vec{r}^{-}}$ are obtained using equations 2.12 and 2.13

$$V^{+}\overline{\vec{r}^{+}} = \int \frac{(u^{-} - u^{+})}{2}\vec{n}^{+}dA \tag{2.12}$$

$$V^{-}\overline{\vec{r}^{-}} = \int \frac{(u^{+} - u^{-})}{2}\vec{n}^{-}dA \tag{2.13}$$

where + and – refer to the current cell and the neighboring cell, V is the volume of the SV. Given the CV averaged lifting functions, $\overline{\vec{r}^{+}}$ and $\overline{\vec{r}^{-}}$, m^{th} degree polynomials can be constructed such that they are $(m+1)^{th}$ order approximation to the CV averaged lifting functions. These polynomials are the actual lifting functions $\vec{r}^{+}$ and $\vec{r}^{-}$. The gradients used for the SV boundary faces are given by

$$\underline{q} = \frac{1}{2}(q_R + \vec{r}^{+} + q_L + \vec{r}^{-}) \tag{2.14}$$

B. Analysis of LDG, the penalty formulation and BR2 methods

B1. One dimensional analysis

In this analysis, we follow a technique described by Zhang and Shu [29] and focus on linear, quadratic and cubic reconstructions. The SV is partitioned into two equal CVs for the second order, three CVs for the third order and four CVs for the fourth order. For the sake of simplicity, let us first consider a linear partition (shown in figure 2). In this case, the formulations can be cast in the following form:

$$\frac{d}{dt}\begin{bmatrix}\bar{u}_{j,1}\\ \bar{u}_{j,2}\end{bmatrix} = \mathrm{A}\begin{bmatrix}\bar{u}_{j-2,1}\\ \bar{u}_{j-2,2}\end{bmatrix} + \mathrm{B}\begin{bmatrix}\bar{u}_{j-1,1}\\ \bar{u}_{j-1,2}\end{bmatrix} + \mathrm{C}\begin{bmatrix}\bar{u}_{j,1}\\ \bar{u}_{j,2}\end{bmatrix} + \mathrm{D}\begin{bmatrix}\bar{u}_{j+1,1}\\ \bar{u}_{j+1,2}\end{bmatrix} + \mathrm{E}\begin{bmatrix}\bar{u}_{j+2,1}\\ \bar{u}_{j+2,2}\end{bmatrix} \tag{2.15}$$

where A,B,C,D and E are constant matrices. We now seek general solution of the following form

$$u(x,t) = \hat{u}_k(t)e^{ikx} \tag{2.16}$$

where k is the index of modes (k=1,2..) representing the wave number. Obviously, the analytical solution for equation 2.1 is $u(x,t) = e^{ikx - k^2 t}$. The solution we are looking for can be expressed as

$$\begin{bmatrix}\bar{u}_{j,1}\\ \bar{u}_{j,2}\end{bmatrix} = \begin{bmatrix}\hat{u}_{k,1}\\ \hat{u}_{k,2}\end{bmatrix} e^{ikx_{j,3/2}} \tag{2.17}$$

Substituting 2.17 into 2.15, we obtain the advancement equation:

$$\begin{bmatrix} \hat{u}'_{k,1} \\ \hat{u}'_{k,2} \end{bmatrix} = G(k,h) \begin{bmatrix} \hat{u}_{k,1} \\ \hat{u}_{k,2} \end{bmatrix} \tag{2.18}$$

Where the amplification matrix is given by

$$G = e^{-2ikh} A + e^{-ikh} B + C + e^{ikh} D + e^{2ikh} E \tag{2.19}$$

The above method can be easily extended to 3rd and 4th orders. In general, all but one of the Eigen values of G is made up of spurious modes and is damped rapidly. The error associated with the scheme, convergence properties can be determined by analyzing the non spurious mode

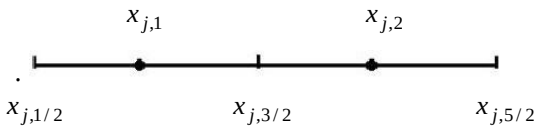


Figure 2. Linear spectral volume in 1D

B1.1 Spatial Analysis

Figure 3 shows the variation of the principal Eigen value with respect to the non dimensional frequency χ (=kh) for the 2nd order case. This is also referred to as the Fourier footprint. Figure 4 shows the error at lower wave numbers. It is clear that the errors associated with the penalty formulation are lower than that of LDG. Moreover, the error associated with the penalty scheme is lower than the LDG error. This means that the system is damped faster in the penalty case and hence converges faster. The situation is similar for the 3rd order as is seen from figures 5 and 6. Thus in general, we expect the penalty scheme to converge faster than the LDG for the 2nd and 3rd orders. The principal Eigen value lies on top of the $y=-x^2$ parabola for the fourth order (figure 7). The fourth order errors are too negligible (figure 8) to comment on convergence phenomena. It must be noted that the matrices A and E are zero matrices for the LDG and BR2 formulations.

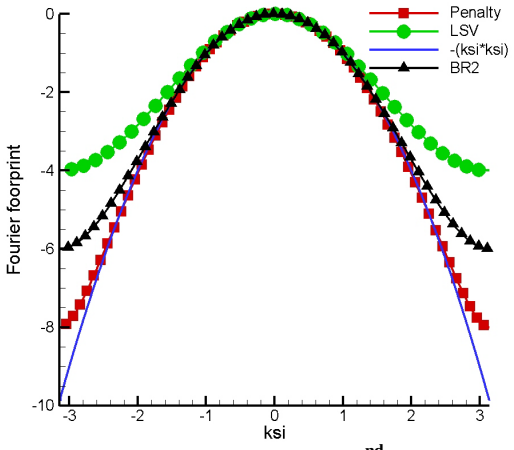


Figure 3. Fourier footprint for 2nd order scheme

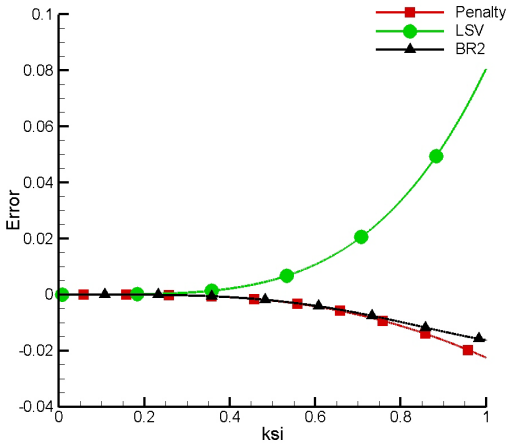


Figure 4. Error associated at low wavenumbers for the 2nd order scheme

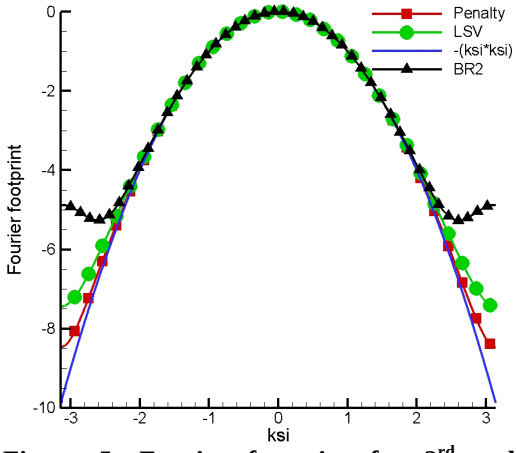


Figure 5. Fourier footprint for 3rd order scheme

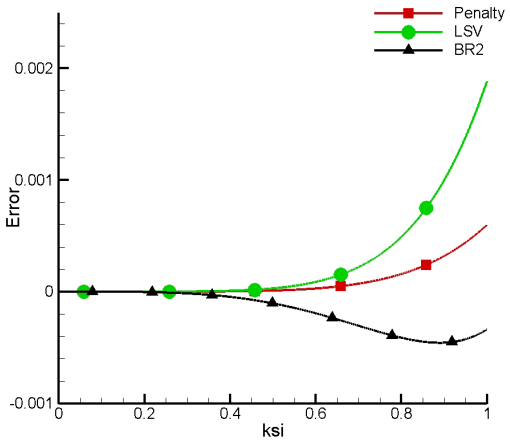


Figure 6. Error associated at low wavenumbers for the 3rd order scheme

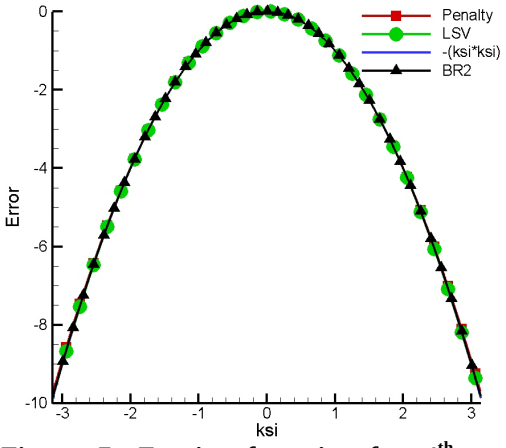


Figure 7. Fourier footprint for 4th order scheme

B1.2 Temporal Analysis

In this section, we compute the time step requirements of the 3-stage Runge Kutta scheme. Table 1 lists the maximum non-dimensional time step ($\tau = \mu \frac{dt}{dx^2}$) required for obtaining a stable solution. It can be seen that the penalty and BR2 methods permit a higher time-step limit than the LDG method. In fact, the time step permitted by the BR2 method is more than double of that of the penalty method. Numerical experiments verified the above conclusion.

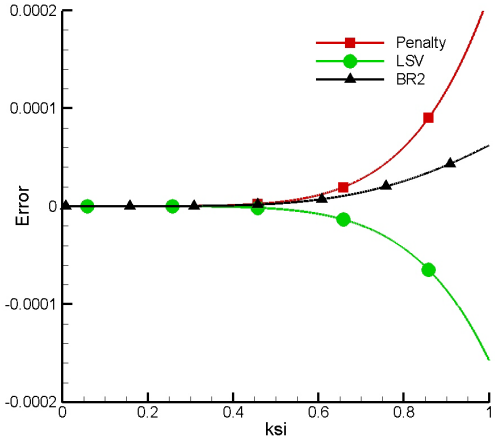


Figure 8. Error associated at low wavenumbers for the 4th order scheme

Table 1. Maximum non-dimensional time step for obtaining stable solutions for the LDG, penalty and BR2 methods.

Case	2nd order (stability)	3rd order (stability)	4th order (stability)
LDG	0.157	0.0266	0.0109
Penalty	0.182	0.0322	0.0133
BR2	0.314	0.0784	0.0321

B2. Two dimensional analysis

In this analysis, we follow a technique described by Abeele et al [30] and focus on the linear reconstruction. The SV is partitioned into 3 CVs. We had to use a basic unit, for imposing periodicity. In a one-dimensional case, the basic unit was the spectral volume itself. It comprises of 2 SVs in 2D. The basic unit (only SVs are shown) is shown in figure 9. This solution process would result in solving a [6*6] set of equations. The rate of change of the solution (in the basic unit) can be written as a linear combination of solutions from the left (L), right(R), top (T), bottom (B) and central(C) units. The formulations can be cast in the following form:

$$\frac{d}{dt}\left[\bar{u}_{i,j}\right] = \mathrm{L}\left[\bar{u}_{i-1,j}\right] + \mathrm{R}\left[\bar{u}_{i+1,j}\right] + \mathrm{C}\left[\bar{u}_{i,j}\right] + \mathrm{B}\left[\bar{u}_{i,j-1}\right] + \mathrm{T}\left[\bar{u}_{i,j+1}\right] \qquad (2.20)$$

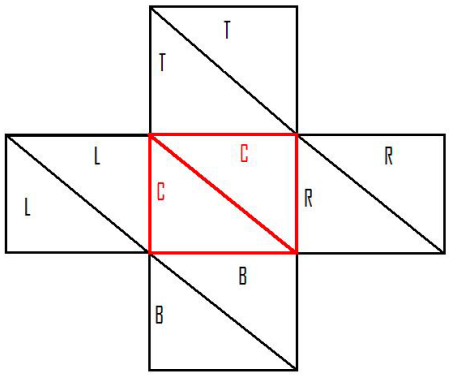


Figure 9. Basic Building block colored in red; the neighbors are used in analysis

where L,R,C,B and T are constant matrices and $\left[\bar{u}_{i,j}\right]$ is the solution vector of the basic building unit. We now seek general solution of the following form

$$u(x,t) = \hat{u}_k(t)e^{ik(x\cos(\theta)+y\sin(\theta))} \tag{2.21}$$

where k is the index of modes (k=1,2..) representing the wave number and θ is the direction of wave propagation. Obviously, the analytical solution for equation 2.1 is $u(x,t) = e^{ikx-k^2t}$. We follow the method described in the 1D analysis section and obtain the following result:

$$\left[\hat{u}_k'\right] = G(k,h)\ \left[\hat{u}_k\right] \tag{2.22}$$

Where the amplification matrix is given by

$$G = e^{-ikh\cos(\theta)}\,L + e^{ikh\cos(\theta)}\,R + C + e^{-ikh\sin(\theta)}\,B + e^{ikh\sin(\theta)}\,T \tag{2.23}$$

B2.1 Spatial Analysis

There is an extra degree of freedom, θ. Due to symmetry of the system and of the building unit, we need to focus on θ varying from 0 to π/4 radians. In this paper, we focus on these extreme cases. Figures 10 and 11 show the variation of the principal Eigen value with respect to the non dimensional frequency χ (=kh) for θ = 0 and π/4 radians respectively. Figures 12 and 13 show the errors at low frequencies. In general, the trends are qualitatively similar to those obtained using 1D analysis. In addition, it can be seen that the errors are much lower when θ = π/4 radians. This is probably due to the way our SVs are defined (right triangles).

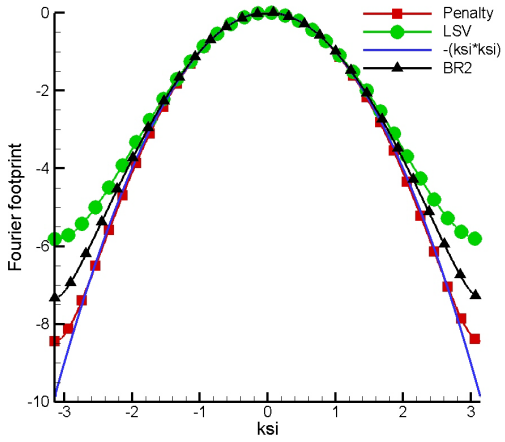


Figure 10. Fourier footprint for the 2D 2nd order scheme for θ = 0 radians

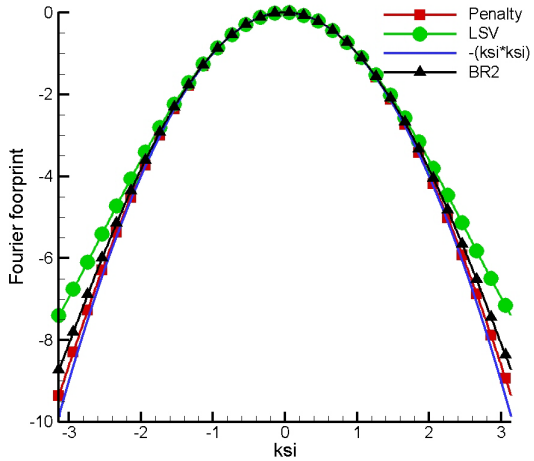


Figure 11. Fourier footprint for the 2D 2nd order scheme for θ = π/4 radians

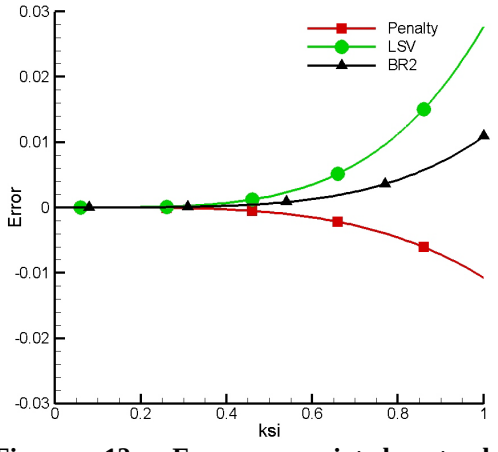


Figure 12. Error associated at low wavenumbers for the 2D 2nd order scheme for θ = 0 radians

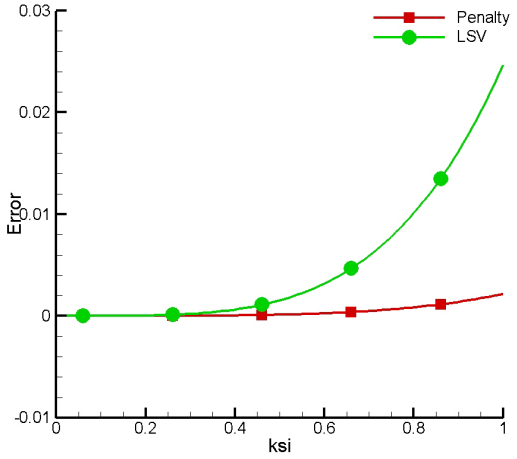


Figure 13. Error associated at low wavenumbers for the 2D 2nd order scheme for θ = = π/4 radians

B2.2 Temporal Analysis

As in the 1D analysis, we stick to the 3-stage SSP Runge Kutta scheme. Table 2 lists the maximum non dimensional time step required for obtaining a stable solution. Once again, the penalty and BR2 methods permit a higher time-step limit than the LDG method. And the time step permitted by the BR2 method is more than that of the penalty method.

Table 2. Maximum non-dimensional time step for for obtaining stable solutions for the 2nd order, 2D LDG, penalty and BR2 methods

Case	θ = 0	θ = π/4
LDG	0.0267	0.0267
Penalty	0.0465	0.0465
BR2	0.0635	0.0635

Implicit LU-SGS formulation for the Spectral volume formulation

In order to solve the flow to a steady state from a nearly arbitrary initial guess, a relaxation scheme is needed. The commonly used time integration scheme is the explicit multi-stage Runge Kutta scheme. The main advantage of the Runge Kutta scheme is that it requires little memory for storage. In addition, this method is inherently simple and so is easy to program. These are the main reasons for it being one of the most preferred methods of time integration.

The main bottleneck associated with the Runge Kutta scheme is the limit imposed on the time step. Euler (and Navier Stokes) equations for realistic geometries entail a rather strict limit on the time step. Even though the above can be circumvented by using a very high order (several RK steps) scheme, it is seldom used as it required lots of storage and thus adversely affects its simplistic nature. Therefore, it is inevitable to use an implicit scheme.

In this paper, we use an implicit LU-SGS scheme for the temporal integration. This scheme has been in use for solving higher order methods. Consider the equation given below

$$\frac{\partial \overline{Q}}{\partial t} = R(\overline{Q}) \tag{3.1}$$

where $\overline{Q}$ is the cell averaged solution vector of a spectral volume. At each current cell *c*, using the backward Euler difference, (3.1) can be written as

$$\frac{\overline{Q}_c^{n+1} - \overline{Q}_c^{n}}{\Delta t} - \left[R_c(\overline{Q}^{n+1}) - R_c(\overline{Q}^{n}) \right] = R_c(\overline{Q}^{n}) \tag{3.2}$$

Let $\Delta \overline{Q}_c = \overline{Q}_c^{n+1} - \overline{Q}_c^{n}$ and linearizing the residual, we obtain

$$R_c(\overline{Q}^{n+1}) - R_c(\overline{Q}^{n}) \approx \frac{\partial R_c}{\partial \overline{Q}_c} \Delta \overline{Q}_c + \sum_{nb \neq c} \frac{\partial R_c}{\partial \overline{Q}_{nb}} \Delta \overline{Q}_{nb}, \tag{3.3}$$

where *nb* indicates all the neighbouring cells contributing to the residual of cell *c*. Therefore, the fully linearized equations for (3.2) can be written as

$$\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial \overline{Q}_c}\right)\Delta\overline{Q}_c-\sum_{nb\neq c}\frac{\partial R_c}{\partial \overline{Q}_{nb}}\Delta\overline{Q}_{nb}=R_c(\overline{Q}^n). \tag{3.4}$$

However, it costs too much memory to store the left-hand side implicit Jacobian matrices. Therefore, we employ a LU-SGS scheme to solve (3.4), and it utilizes the most recent solution for the neighbouring cells,

$$\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial \overline{Q}_c}\right)\Delta\overline{Q}_c^{(k+1)}=R_c(\overline{Q}^n)+\sum_{nb\neq c}\frac{\partial R_c}{\partial \overline{Q}_{nb}}\Delta\overline{Q}_{nb}^{*}. \tag{3.5}$$

The Jacobian matrix

$$D=\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial \overline{Q}_c}\right), \tag{3.6}$$

is the element (or cell) matrix, which is not very big for 2nd to 3rd order SV schemes. For instance, D is [6x6] for the 2D 3rd order SV wave equation. Neither do we want to store the matrices $\frac{\partial R_c}{\partial \overline{Q}_{nb}}$, thus (3.5) is further manipulated as follows

$$\begin{aligned} &R_c(\overline{Q}^n)+\sum_{nb\neq c}\frac{\partial R_c}{\partial \overline{Q}_{nb}}\Delta\overline{Q}_{nb}^{*}=R_c(\overline{Q}_c^n,\{\overline{Q}_{nb}^n\})+\sum_{nb\neq c}\frac{\partial R_c}{\partial \overline{Q}_{nb}}\Delta\overline{Q}_{nb}^{*}\\ &\approx R_c(\overline{Q}_c^n,\{\overline{Q}_{nb}^{*}\})\approx R_c(\overline{Q}_c^{*},\{\overline{Q}_{nb}^{*}\})-\frac{\partial R_c}{\partial \overline{Q}_c}\Delta\overline{Q}_c^{*}=R_c(\overline{Q}^{*})-\frac{\partial R_c}{\partial \overline{Q}_c}\Delta\overline{Q}_c^{*}. \end{aligned} \tag{3.7}$$

Let $\Delta^{*}\overline{Q}_c^{(k+1)}=\Delta\,\overline{Q}_c^{(k+1)}-\Delta\,\overline{Q}_c^{*}$. We can combine (3.5) and (3.7) together to obtain

$$\left(\frac{I}{\Delta t}-\frac{\partial R_c}{\partial \overline{Q}_c}\right)\Delta^{*}\overline{Q}_c^{(k+1)}=R_c(\overline{Q}^{*})-\frac{\Delta\overline{Q}_c^{*}}{\Delta t}. \tag{3.8}$$

Equation (3.8) is then solved with a direct LU decomposition solver for an element and the solver is swept symmetrically forward and backward through all the computational grid elements. Once (3.8) is solved to machine zero, the unsteady residual is zero at each time step. For steady flow problem, we found that the term $\frac{\Delta\overline{Q}_c^{*}}{\Delta t}$ in the right-hand-side of (3.8) can be neglected and leading to quicker convergence. Note that $\Delta\overline{Q}_c^{*}$ is the difference between the current solution $\overline{Q}_c^{*}$ and the solution at the previous time level $\overline{Q}_c^n$. In reality, the entire system is swept several times in order to proceed to the next time level. As a result, $\Delta\overline{Q}_c^{*}$ is influenced by the solution occurred several sweeps ago. This introduces an under-relaxation effect. Hence, neglecting the $\frac{\Delta\overline{Q}_c^{*}}{\Delta t}$ term accelerates the convergence.

p-Multigrid Method

The Gauss-Seidel or Jacobi iterations produce smooth errors when applied on the above mentioned equations. The error vector has its high frequencies nearly removed in a few iterations using a higher order polynomial; but low frequencies are removed very slowly. The key idea of the p-Multigrid method is to solve the nonlinear equations

using a lower order polynomial such that "smooth becomes rough" and low frequencies act like high frequencies. Such a p-Multigrid method has been used for high-order discontinuous Galerkin method; see [31]. The p-Multigrid method operates on a sequence of solution approximations of different polynomial orders. Therefore it offers the flexibility of switching between higher and lower polynomial levels without changing the actual geometrical nodal grid points.

To accomplish the communication between different levels, restriction (I_p^{p-1}, I_{p-1}^{p-2}) and prolongation ($I_{p-1}^{p}, I_{p-2}^{p-1}$) operators are required in addition to the aforementioned relaxation scheme as a smoother. Restriction consists of moving solutions and their residuals at the unknown points from a space of higher polynomial order to a lower polynomial order. Prolongation refers to a reverse procedure in which lower order polynomial solution correction is redistributed as corrections to the solutions of the unknown points at a higher polynomial order.

Classical multigrid method begins with a two-level process. First, iterative relaxation is applied using the higher order polynomial, thus basically reducing high-frequency errors. Then, a "coarse-grid" correction is applied, in which the smooth error is determined at the lower polynomial level. This error is interpolated to the higher polynomial level and used to correct the existing higher order solutions. Applying this method recursively to solve the lower polynomial level problems leads to multigrid.

Defining three polynomial levels from the highest to the lowest as *p*, *p-1* and *p-2*, we want to solve:

$$R_p(\overline{Q}_p) = r_p\,, \tag{4.1}$$

and the rhs r_p is zero for the highest level polynomial;

$$R_{p-1}(\overline{Q}_{p-1}) = r_{p-1}\ ; \tag{4.2}$$

$$R_{p-2}(\overline{Q}_{p-2}) = r_{p-2}\,. \tag{4.3}$$

We want to employ the implicit LU-SGS schemes as the smoothers for all three levels. For simplicity, the following steps summarise a standard two-grid scheme with a V cycle at p and p-1 levels.

- Improve $\overline{Q}_p$ by application of a few times the smoother similar as equation (3.8)

$$\left[\frac{I}{\Delta t} - \frac{\partial R_c^p}{\partial \overline{Q}_c^p}\right]^n \Delta^2 \overline{Q}_c^p = R_p(\overline{Q}_p^*) - r_p \tag{4.4}$$

- Restrict the latest solution $\overline{Q}_p$ to the coarser level for an approximate solution $\overline{Q}_{p-1}$

$$\overline{Q}_{p-1}^0 = I_p^{p-1}\left(\overline{Q}_p\right) \tag{4.5}$$

- Compute the defect on the finest level

$$d_p = r_p - R_p\left(\overline{Q}_p\right) = -R_p\left(\overline{Q}_p\right) \tag{4.6}$$

- Compute the right hand side of equation (4.2) as

$$r_{p-1} = R_{p-1}\left(\overline{Q}_{p-1}^0\right) + I_p^{p-1} d_p \tag{4.7}$$

- Improve $\overline{Q}_{p-1}$ by application of a few times the smoother

$$\left[\frac{I}{\Delta t} - \frac{\partial R_c^{p-1}}{\partial \overline{Q}_c^{p-1}}\right]^n \Delta^2 \overline{Q}_c^{p-1} = R_{p-1}(\overline{Q}_{p-1}^*) - r_{p-1} \tag{4.8}$$

- Correct the current solution on the finest level by

$$\overline{Q}_p = \overline{Q}_p + I_{p-1}^p(\overline{Q}_{p-1} - \overline{Q}_{p-1}^0) \tag{4.9}$$

- Improve $\overline{Q}_p$ by application of a few times the iterative smoother to get $\overline{Q}_p$

We can call the above two-grid V cycle recursively for p-1 and p-2 levels just behind the smoother (equation 4.8) to accomplish a three-level V cycle. In fact, if it is called twice, a W cycle is constructed

Test Results

In this Section, numerical experiments are conducted using the penalty method. The following two cases are presented.

A. 2D diffusion equation

A1. Accuracy study

In this section, we test the accuracy of the penalty scheme using the 2D diffusion equation:

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial x^2} - \frac{\partial^2 u}{\partial y^2} = 0\,, \qquad (x, y) \in [0,1]^2 \tag{5.1}$$

$$u_{analytical} = \sin(x)e^{y} \tag{5.2}$$

The steady state analytical solutions are specified at all the boundaries.

In tables 3, 4 and 5, we present the SV averaged L1 and Linf errors for the 2nd, 3rd and 4th orders respectively using the LDG, penalty and BR2 formulations. It can be seen that the 2nd and the 4th order schemes reach their respective orders asymptotically for all the schemes. We do experience a drop in order for the 3rd order (odd ordered) scheme. This is under investigation. It can be observed that the LDG method produces the maximum errors in general. Regular grids were used for the study (figure 14). The solution obtained using a 4th order stencil on a 40x40x2 grid is shown in figure 15.

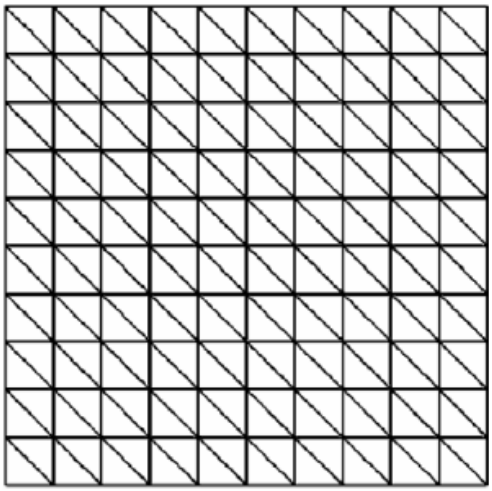


Figure 14. regular grid(10x10x2) used for validation study

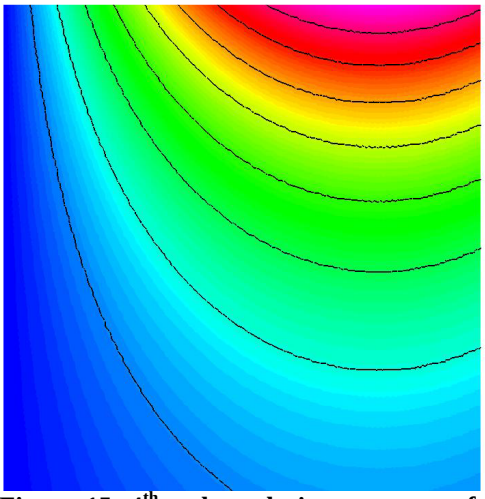


Figure 15. 4th order solution contours of the equation 5.1 using a 40x40x2 grid

Table 3. Accuracy of 2nd order 2D diffusion equation on regular grids

Grid	L1 error (Penalty)	L1 order (Penalty)	Linf error (Penalty)	Linf order (Penalty)
10x10x2	1.72e-03	-	5.99e-03	-
20x20x2	4.40e-04	1.97	1.52e-03	1.98
40x40x2	1.12e-04	1.97	3.85e-04	1.98
80x80x2	2.83e-05	1.99	9.71e-05	1.99
Grid	L1 error	L1 order	Linf error	Linf order

	(LDG)	(LDG)	(LDG)	(LDG)
10x10x2	4.95e-03	-	1.41e-02	-
20x20x2	1.26e-03	1.98	3.72e-03	1.95
40x40x2	3.48e-04	1.90	9.78e-04	1.95
80x80x2	1.28e-04	1.65	2.63e-04	1.93
Grid	L1 error (BR2)	L1 order (BR2)	Linf error (BR2)	Linf order (BR2)
10x10x2	2.28e-03		1.20e-02	
20x20x2	4.90e-04	2.22	3.19e-03	1.91
40x40x2	1.12e-04	2.13	8.19e-04	1.96
80x80x2	2.65e-05	2.07	2.07e-04	1.98

Table 4. Accuracy of 3rd order 2D diffusion equation on regular grids

Grid	L1 error (Penalty)	L1 order (Penalty)	Linf error (Penalty)	Linf order (Penalty)
10x10x2	8.67e-05	-	3.19e-04	-
20x20x2	1.43e-05	2.6	4.66e-05	2.78
40x40x2	2.66e-06	2.43	6.39e-06	2.86
80x80x2	5.53e-07	2.27	1.23e-06	2.38
Grid	L1 error (LDG)	L1 order (LDG)	Linf error (LDG)	Linf order (LDG)
10x10x2	1.27e-04	-	2.81e-04	-
20x20x2	2.43e-05	2.38	5.41e-05	2.37
40x40x2	5.44e-06	2.16	1.27e-05	2.09
80x80x2	1.23e-06	2.10	2.96e-06	2.07
Grid	L1 error (BR2)	L1 order (BR2)	Linf error (BR2)	Linf order (BR2)
10x10x2	9.49e-05	-	2.68e-04	-
20x20x2	1.31e-05	2.85	3.80e-05	2.82
40x40x2	2.14e-06	2.61	5.21e-06	2.88
80x80x2	3.90e-07	2.46	9.71e-07	2.42

Table 5. Accuracy of 4th order 2D diffusion equation on regular grids

Grid	L1 error (Penalty)	L1 order (Penalty)	Linf error (Penalty)	Linf order (Penalty)
10x10x2	3.69e-06	-	9.56e-06	-
20x20x2	2.35e-07	3.97	6.01e-07	3.99
40x40x2	1.49e-08	3.98	3.74e-08	4.00
Grid	L1 error (LDG)	L1 order (LDG)	Linf error (LDG)	Linf order (LDG)
10x10x2	4.30e-06	-	9.74e-06	-
20x20x2	2.97e-07	3.8	6.74e-07	3.8
40x40x2	2.06e-08	3.9	4.43e-08	3.9
Grid	L1 error (BR2)	L1 order (BR2)	Linf error (BR2)	Linf order (BR2)
10x10x2	4.17e-06	-	9.99e-06	-
20x20x2	2.76e-07	3.91	6.68e-07	3.90
40x40x2	1.78e-08	3.95	4.28e-08	3.95

A2. Convergence study

In this section, we perform numerical experiments to study the effect of the implicit LU-SGS smoother and the p-multigrid algorithm. The comparison of the convergence speeds between the implicit and the explicit Runge Kutta schemes are shown in Figures 16, 17 and 18 for second, third and fourth orders respectively. It can be seen that the speedup increases with increasing the order of the interpolation polynomial. We attain a speedup of nearly 20 for 4^{th} order simulations. Thus it is more practical to use implicit algorithms for higher order methods. Figures 19 and 20 show the effect of the p-multigrid on the 3^{rd} and 4^{th} orders respectively. It can be seen that the multigrid results in a speed-up of 3 for the 3^{rd} order and nearly 7 for the 4^{th} order. Thus a combination of implicit LU-SGS and p-multigrid results in a speedup which is more than 2 orders in magnitude!

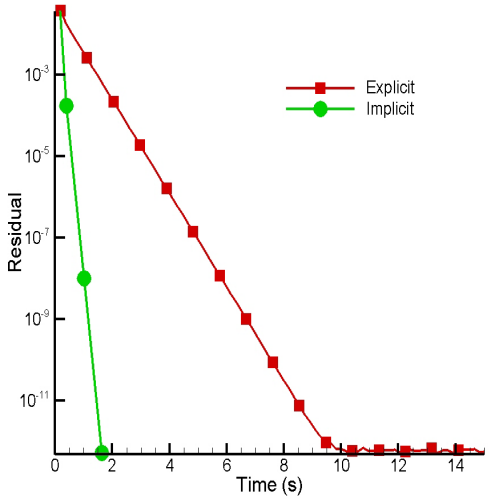


Figure 16. Convergence history using explicit R-K and implicit LU-SGS schemes for the 2D diffusion equation using a 2^{nd} order scheme

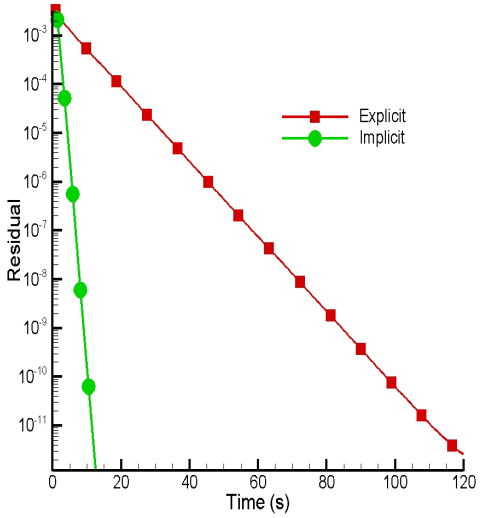


Figure 17. Convergence history using explicit R-K and implicit LU-SGS schemes for the 2D diffusion equation using a 3rd order scheme

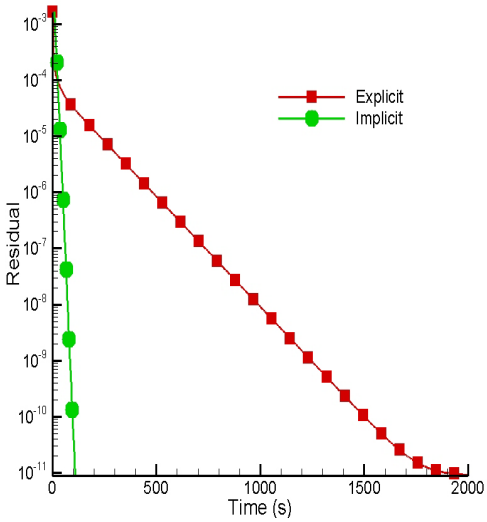


Figure 18. Convergence history using explicit R-K and implicit LU-SGS schemes for the 2D diffusion equation using a 4th order scheme

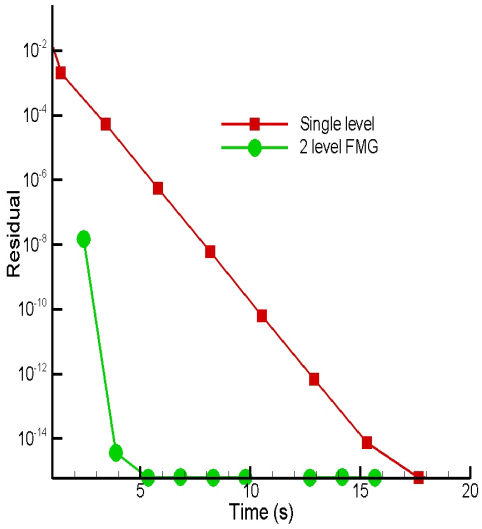


Figure 19. Convergence history using single level and 2 level FMG LU-SGS schemes for the 3rd order 2D diffusion equation.

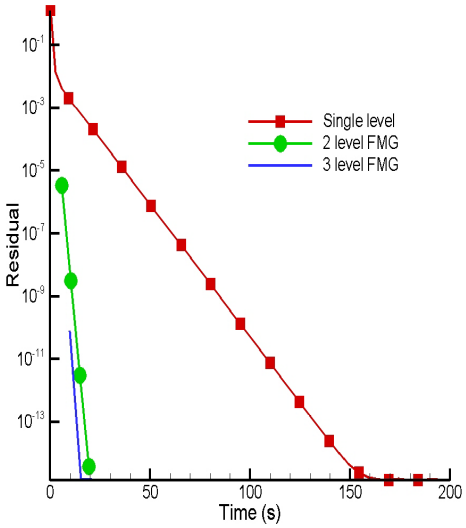


Figure 20. Convergence history using single level, 2 level FMG and 3 level FMG LU-SGS schemes for the 4th order 2D diffusion equation.

B. 2D Navier Stokes equations

We were able to extend the viscous formulation for Navier stokes equations. The inviscid fluxes were handled using an approximate Riemann solver (Rusanov flux). The penalty method was used for the viscous fluxes.

A1. Flow over an airfoil

In this section, we simulate flow over a naca0012 airfoil. The flow was transonic at Mach 0.5 and had a zero angle of attack. The Reynolds number was 1000. Figures 21, 22 and 23 show the convergence histories for 2nd, 3rd and 4th orders respectively. It can be seen that the speedup at higher orders is massive (more than 2 orders), especially for 4th order. The 3rd and 4th order explicit schemes limit the CFL to a few hundredths. In contrast, the implicit schemes have a CFL limit of more than a hundred. This is one the main reasons for obtaining the above mentioned speedups.

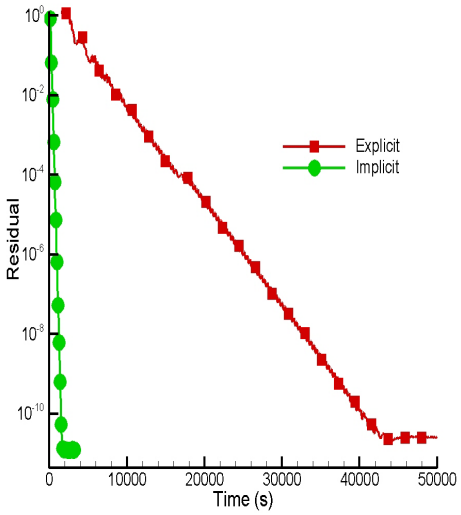


Figure 21. Convergence history using explicit R-K and implicit LU-SGS schemes for the flow over naca0012 airfoil using a 2nd order scheme

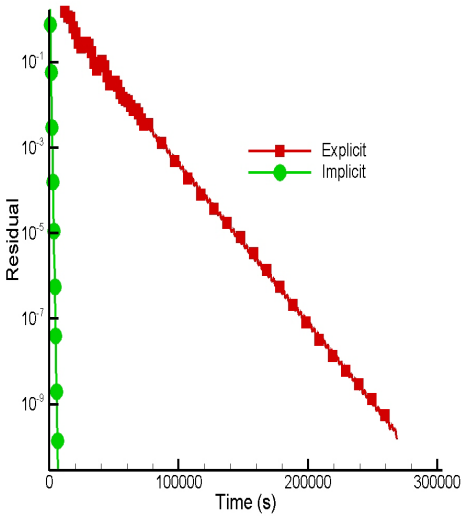


Figure 22. Convergence history using explicit R-K and implicit LU-SGS schemes for the flow over naca0012 airfoil using a 3rd order scheme

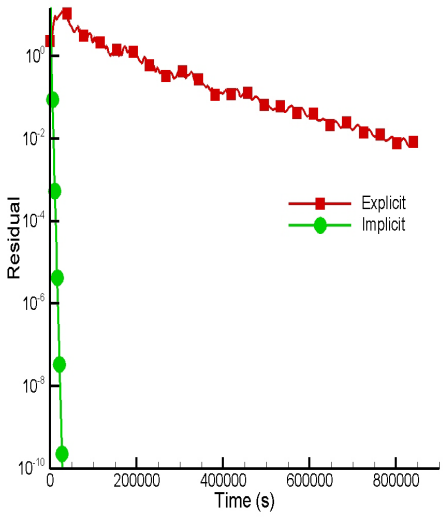


Figure 23. Convergence history using explicit R-K and implicit LU-SGS schemes for the flow over naca0012 airfoil using a 4th order scheme

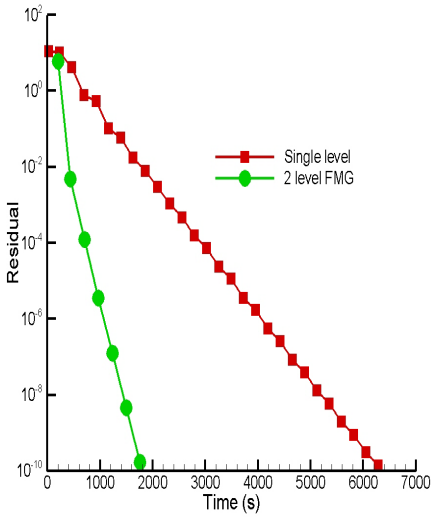


Figure 24. Convergence history using single level and 2 level FMG LU-SGS schemes for the flow over naca0012 airfoil using a 3rd order scheme

Figure 24 shows the effect of p-multigrid for the 3rd order simulation. The lower level operations were carried out using a larger time step. This was done to eliminate the low frequency errors at a faster rate. Multigrid for the 4th order was found to be unstable and is under investigation. Fig. 25 shows the Mach contours computed with linear and quadratic SVs. The solution gets smoother and smoother with the increasing of the order of the polynomial reconstruction. The wake region looks more refined and continuous for the 3rd order case. The difference between the 3rd and the 4th order plots were negligible.

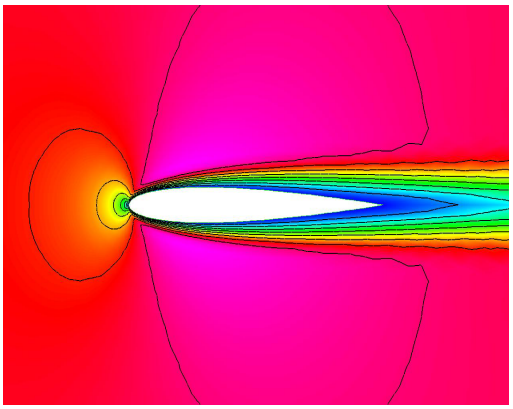
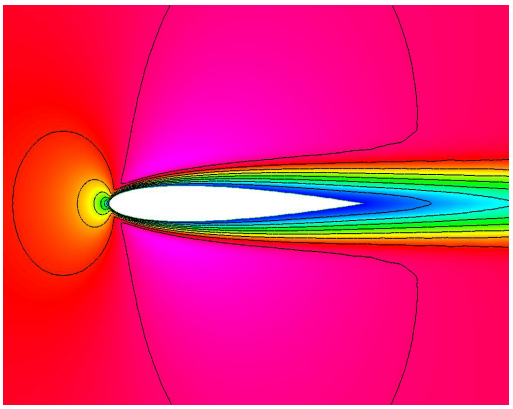
(a)  (b) 

Figure 25. Mach contours around the NACA0012 airfoil at zero degree of attack, Re=1000, M=0.5. (a) 2ndorder; (b) 3rd order

A2. Flow over an flat plate

In this section, we simulate laminar flow over a flat plate. naca0012 airfoil. The flow was transonic at Mach 0.5. Figures 26, 27 and 28 show the convergence histories for 2nd, 3rd and 4th orders respectively. Figure 29 shows the effect of p-multigrid for the 3rd order simulation. These convergence plots look qualitatively similar to those obtained for the flow over naca0012 airfoil. It must be noted that the speedups obtained for this simulation are slightly lower than those obtained for the airfoil case. This is because of the surface grid: airfoil utilizes a curved boundary for the 3rd and 4th orders. This makes the system stiffer and the effect of implicit/multigrid schemes result in enormous speedups.

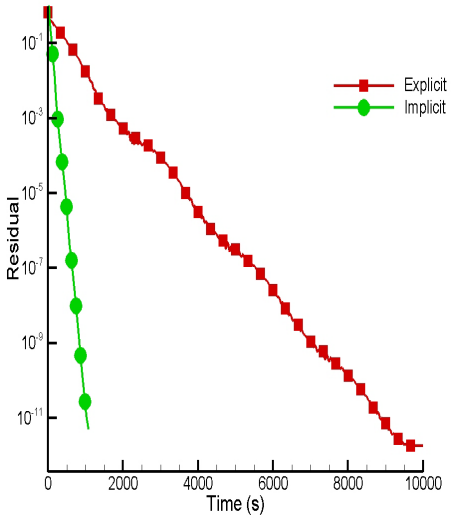


Figure 26. Convergence history using explicit R-K and implicit LU-SGS schemes for the flow over a flatplate using a 2nd order scheme

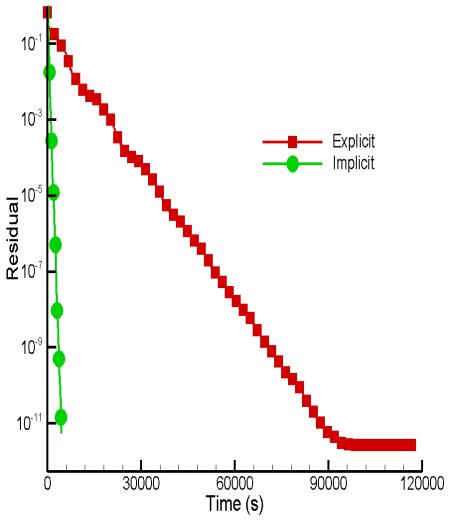


Figure 27. Convergence history using explicit R-K and implicit LU-SGS schemes for the flow over a flatplate using a 3rd order scheme

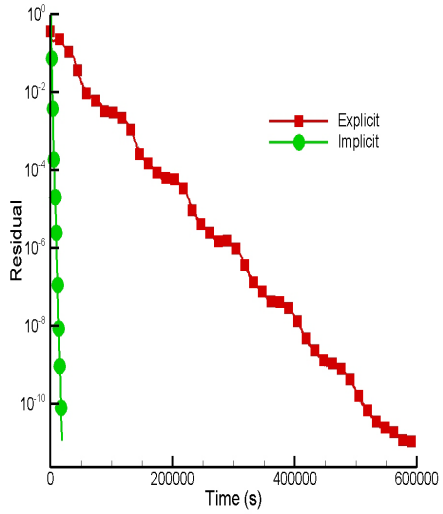


Figure 28. Convergence history using explicit R-K and implicit LU-SGS schemes for the flow over a flatplate using a 4th order scheme

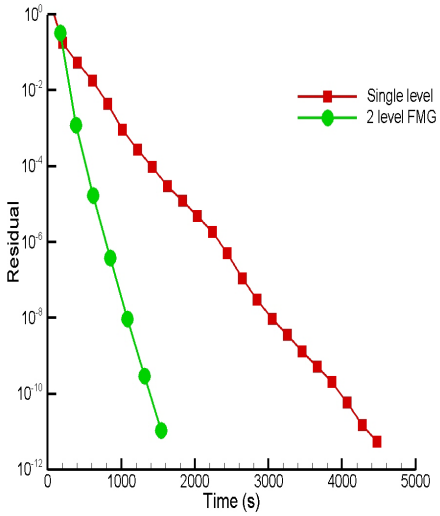


Figure 29. Convergence history using single level and 2 level FMG LU-SGS schemes for the flow over a flatplate using a 3rd order scheme

Conclusions

In the present study, we suggested a new penalty approach and implemented the BR2 method for computing viscous fluxes. Fourier analysis was done for both the new formulations and the previously used LDG scheme. The results of the Fourier analysis implied that the new methods converge faster, have better accuracy and can handle bigger time steps (explicit simulations) than the LDG method. In particular, the BR2 method was able to handle much larger time steps than the penalty approach. The accuracy of the three methods was tested by performing an accuracy study using the 2D Laplace equation.

We also incorporated the implicit LU-SGS scheme into the SV method. This resulted in extraordinary speedups. A p-multigrid was also implemented for the diffusion and the NS equations. The implicit solver and the p-multigrid were successfully coupled in order to enable efficient simulation. A combination of the above gave speedups of more than 2 orders of magnitude. Future research includes the following areas

1.Perform some analysis on the p-multigrid method.
2. Test the 2nd formulation of Bassi and Rebay for Navier-Stokes equations.
3.Some temporal analysis for the implicit schemes.

Acknowledgements

The authors gratefully acknowledge the support from the Air Force Office of Scientific Research (Grant Number FA9550-04-1-0053), with Dr. Fariba Fahroo as the Technical Monitor.

References

1. Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210.
2. Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation, J. Comput. Phys. 179 (2002) 665.
3. Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids III: extension to one-dimensional systems, J. Sci. Comput. 20 (2004) 137.

4. Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations, J. Comput. Phys. 194 (2004) 716.
5. Z.J. Wang, Y. Liu, Extension of the spectral volume method to high-order boundary representation, J. Comput. Phys. 211 (2006) 154–178.
6. T.J. Barth, P.O. Frederickson, High-order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. 90-0013, 1990.
7. M. Delanaye, Yen Liu, Quadratic reconstruction finite volume schemes on 3D arbitrary unstructured polyhedral grids, AIAA Paper No. 99-3259-CP, 1999.
8. A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order essentially non-oscillatory schemes III, J. Comput. Phys. 71 (1987) 231.
9. R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: analysis and implementation, J. Comput. Phys. 114 (1994) 45–58.
10. A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49, 357(1983)
11. Y. Sun, Z.J. Wang and Y. Liu, "Spectral (finite) volume method for conservation laws on unstructured grids VI: Extension to viscous flow", J. Comput. Phys.2006; 215:41-58.
12. B. Cockburn, C.-W. Shu, The local discontinuous Galerkin method for time-dependent convection diffusion system, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
13. F. Bassi, S. Rebay, "A high-order accurate discontinuous finite element method for the numerical solution of the compressible Navier Stokes equations", J. Comput. Phys., Vol. 131, pp. 267-279, 1997.
14. A. Jameson, S. Yoon, Lower–upper implicit schemes with multiples grids for the Euler equations, AIAA Journal 25 (7) (1987) 929–935.
15. D. Caughley, A. Jameson, Fast preconditioned multigrid solution of the Euler and Navier–Stokes equations for steady, compressible flows, International Journal for Numerical methods in Fluids 2003; 43:537-553
16. R.F. Chen and Z.J. Wang, Fast, Block Lower-Upper Symmetric Gauss Seidel Scheme for Arbitrary Grids, *AIAA Journal*, vol. 38, no. 12, pp. 2238-2245, 2000
17. C. Liang, R. Kannan and Z.J. Wang, "A-p-Multigrid Spectral Difference Method with Explicit and Implicit Smoothers on Unstructured Grids", AIAA paper No 2007-4326, 2007
18. E.M. Ronquist, A.T. Patera, Spectral element multigrid, I. Formulation and numerical results, Journal of Scientific Computing 2 (4) (1987) 389–406.
19. Y. Maday, R. Munoz, Spectral element multigrid, Part 2: Theoretical justification, Tech. Rep. 88-73, ICASE, 1988.
20. F. Bassi, S. Rebay, Numerical solution of the Euler equations with a multiorder discontinuous Finite element method, in: Proceedings of the Second International Conference on Computational Fluid Dynamics, Sydney, Australia, 15–19 July 2002.
21. K. J. Fidkowski, T. A. Oliver, J. Lu and D. L. Darmofal, p-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations, Journal of Computational Physics, 207 (2005) pp. 92-113.
22. C. R. Nastase and D. J. Mavriplis, High-order discontinuous Galerkin methods using an hp-multigrid approach, Journal of Computational Physics, 213 (2006) pp. 330-357.
23. H. Luo, J. D. Baum and R. Löhner, A p-multigrid discontinuous Galerkin method for the Euler equations on unstructured grids, Journal of Computational Physics, 211 (2006), pp 767-783.
24. V.V. Rusanov, Calculation of interaction of non-steady shock waves with obstacles, J. Comput. Math. Phys. USSR 1 (1961) 267–279.
25. P.L. Roe, Approximate Riemann solvers, parameter vectors and difference schemes, J. Comput. Phys. 43 (1981) 357–372.
26. M.-S.Liou, and C. Steffen, A New Flux Splitting Scheme, J. Comput. Phys., Vol. 107, 23-39, 1993.
27. F. Brezzi; G. Manzini; D. Marini; P. Pietra; and A. Russo. Discontinuous Galerkin approximations for elliptic problems. Numer. Methods for Partial Differential Eqns, 16:365–378, 2000.
28. F. Bassi and S. Rebay. GMRES discontinuous Galerkin solution of the compressible Navier-Stokes equations. In Karniadakis Cockburn and Shu, editors, Discontinuous Galerkin Methods: Theory, Computation and Applications, pages 197–208. Springer, Berlin, 2000.
29. Zhang M, Shu C-W, An analusis of three different formulations of the discontinuous Galerkin method for diffusion equations, Mathematical Models and Methods in Applied Sciences 2003;13:395-413
30. den Abeele, K. V., Broeckhoven, T., and Lacor, C., "Dispersion and Dissipation Properties of the 1d Spectral Volume Method and Application to a P-Multigrid Algorithm," J. Comp. Phys., Vol. in press., 2007.
31. B. T. Helenbrook and H. L. Atkins, Application of p-Multigrid to Discontinuous Galerkin formulations of the Poisson equation, AIAA Journal, 44 (2006) pp. 566-575.